CLAIMS:

1. A microcontroller (100) the programming of which is carried out in at least one machine-dependent assembler language in which the assembler commands, with the exception of conditional program jumps or program branches, respectively, can be executed in essence independently of data, characterized by at least one random number generator (10) assigned to the microcontroller (100) can be executed, by means of which the program jumps or program branches can be executed

- in dependence on the state of the random number generator (10) and/or
- independently of the internal state of the programming of the microcontroller (100).

2. A microcontroller as claimed in claim 1, characterized by at least one, in particular bit-addressable, random number register (20) assigned to the random number generator (10).

3. A microcontroller as claimed in claim 1 or 2, characterized by an embodiment as a smartcard controller.

4. An electrical or electronic device controlled by means of at least one microcontroller (100) as claimed in at least one of claims 1 to 3.

5. A method for processing the programming of a microcontroller (100) executed in at least one machine-dependent assembler language, the assembler commands, with the exception of conditional program jumps or branches, being executed essentially independently of data, characterized in that the program jumps or program branches are executed

- in dependence on the state of at least one random number generator (10) and/or
- independently of the internal state of the programming of the microcontroller (100).

6. A method as claimed in claim 5, characterized in that the random number generated by the random number generator (10) is read via software via registers and the random number read is then evaluated with a conditional program jump or branch.

7. A method as claimed in claim 5 or 6, characterized in that, if at least one, in particular bit-addressable, random number register (20) is present, testing per bit of the random number register (20) and a conditional jump or branch is carried out.

8. A method as claimed in at least one of claims 5 to 7, characterized by the implementation of at least one assembler command ("branch on random bit"), a defined bit of the random number register (20) being supplied, in particular directly, to the condition input for the conditional jump or branch.

9. A method as claimed in at least one of claims 5 to 8, characterized in that at least one Arithmetic Logic Unit (ALU) flag controlling the conditional jumps or branches is replaced, in particular via the software, by at least one bit of the random number register (20), so that the conditional jumps or branches corresponding to the bit of the Arithmetic Logic Unit are controlled by the bit of the Random Number Register (20).

10. A use of a microcontroller (100) as claimed in at least one of claims 1 to 3 and/or of a method as claimed in at least one of claims 5 to 9 for completely concealing the programming running on the microcontroller (100), so that at least one program running on the microcontroller (100) is unpredictable and non-reproducible for an external observer.